STANDARD PROCUREMENT SYSTEM (SPS)

Continuity of Operations Plan

for

ACTIVITY NAME

DATE

Table of Contents

Executive Summary

The purpose of this Continuity of Operations Plan (COOP) is to describe provisions to ensure that the Standard Procurement System (SPS) support for ***ACTIVITY NAME*** is maintained, and that downtime is minimized. It addresses the technical components of the SPS system, (which uses the Procurement Desktop-Defense - PD^2 software) but does not discuss manual operation of the contracting function in the absence of SPS support.

This document addresses the continuity of operations responsibilities that rest primarily with ***ACTIVITY NAME*** System Administrators (SAs). These roles are an extension of those that are normally required to ensure the proper operation and administration of SPS.

Contracting operations are heavily dependent on automated support, and can continue for only short periods of time using manual methods without serious degradation of contracting capabilities. Near the end of the fiscal year, any downtime is unacceptable. During slower times of the year, scheduled and well-coordinated downtime can span a couple of days. At any time, unscheduled downtime can seriously degrade contracting operations.

This COOP provides specific procedures covering the following three major areas of system maintenance or failure:

- Non-Critical Failure of Hardware, Software or Network Components
- Critical Failure of Critical Hardware, Software or Network Components
- Sudden Power Failure of Unknown Duration

Specific procedures are also provided with respect to day to day backup and recovery operations, and day-to-day maintenance procedures.

1.0 Introduction

The purpose of this Continuity of Operation Plan (COOP) is to describe provisions to ensure that SPS support for contracting functions is maintained, and that downtime is minimized. It addresses procedures related to the technical components of the SPS system, but does not address manual operation of the contracting function in the absence of SPS support.

1.1 System Components

The purpose of this section is to identify those SPS system components that are covered by this COOP. The matrix shown below (Figure 1), lists the components of the SPS system, their hardware platform, and relative priority or importance to proper system operation.

System Component	Description	Hardware Platform	Priority
Database services • Sybase	Provides access to PID 2 data for local and remote users	Database Server	1
Hardware Platforms • Pentium	Host Machines for Database Server	Host machines for Database Server	2
Operating System • NT • Windows 95	Provides operating system services to their respective machines	Database Server	3
Application Services • PD^2 Database • Reference Library • Cognos Catalog	Provides access to the PD^2 server--based systems and resources required by the PD^2 application	Database Server	4
Security Services	Provided by the operating system (OS), database management system (DBMS) and application (Windows 95, New Technology (NT), Sybase and PD^2)	Database Server	5

Figure 1. SPS Recovery Priorities Segmented by Service and Hardware Platform

1.2 Scope

This document addresses the continuity of operations responsibilities that rest primarily with ***ACTIVITY NAME*** SAs. These roles are an extension of those that are normally required to ensure the proper operation and administration of PD^2

1.3 Document Overview

This document provides continuity of operations planning and preparation guidance for the PD^2 system. It identifies potential threats to the proper functioning of PD^2 and identifies provisions to minimize the risk of those threats. This Continuity of Operations Plan (COOP) is divided into the following sections:

- **Section 1** **Introduction.** Describes the purpose and scope of this document.
- **Section 2** **Emergency Response Preparation.** Identifies the key planning considerations related to continuity of PD^2 operations. Identifies provisions for minimizing disruption of service.
- **Section 3** **Emergency Response Procedures.** Identifies procedures for responding to threats to normal PD^2 operations.
- **Section 4** **Backup Operations.** Identifies backup procedures for the database and application servers.
- **Section 5** **Recovery Operations.** Identifies procedures for recovery based on standard backups, and procedures required for bringing backup servers to operational status.
- **Section 6** **Maintenance Procedures.** Describes the purpose and use of a number of day-to-day maintenance procedures, such as those related to monitoring database server usage.

2.0 Emergency Response Preparations

2.1 Operational Continuity Requirements

Contracting operations are heavily dependent on automated support. and can continue for only short periods of time using manual methods without serious degradation of contracting capabilities.

- Near the end of the fiscal year, any downtime is unacceptable.
- During slower times of the year, scheduled and well-coordinated downtime can span a couple of days.
- At any time, unscheduled downtime can seriously degrade contracting operations.

2.2 Software

To ensure that the operational environment can be restored in the event of a failure, copies of installation media and related documentation for each required software product are retained in a readily accessible, secure location.

The software products, version numbers and installation media required for the maintenance and operation of PD^2 are identified in the matrix below. The SAs maintain an inventory of all updated version software and documentation related to PD^2 for ***ACTIVITY NAME*.**

Software Product	Version	Hardware Platform	Medium
Sybase SQL Server	11.02	NT Server	PD^2 Installation CD
PD^2 Database	3.5/4.1	NT Server	PD^2 Installation CD
Windows NT	4.0	NT Server	CD
PD^2 Reference Library	3.5/4.1	NT Server	PD^2 Installation CD
Cognos Catalog	3.5/4.1	NT Server	PD^2 Installation CD
Cognos Impromptu/PowerPlay	4.1	Client Workstation	PD^2 Installation CD
PD^2 Client	3.5/4.1	Client Workstation	PD^2 Installation CD
Microsoft Office 97	97	Client Workstation	CD

2.3 Hardware

2.3.1 Database Server

The server is a multi-CPU machine so failure of a single CPU chip should not bring the server down, but only degrade performance until it can be replaced. In addition, the SAs will establish a maintenance contract to ensure that database server hardware problems of any type receive expert attention within two hours of SAs registering a problem.

2.3.2 Network Infrastructure

The database server is available to users via local and wide area networks. Local area network connections and capabilities are required for print services as well as connections between clients and the servers. Maintenance of these network infrastructure components is not part of the PD^2 program, and is generally the responsibility of ***MAJOR CLAIMANT, BASE, OR ACTIVITY NAME***

2.3.3 Client Workstations and Printers

Maintenance of client workstations and printers is the responsibility of ***ACTIVITY NAME*** SAs .

2.4 Power and Environmental Components

The PD^2 database server is equipped with an uninterruptible power supply (UPS) which provides protection against power failures in addition to power filter/surge protection services. A spare UPS is not a requirement. Recovery from a failed UPS is covered under Section 3.1 on non-critical failure procedures unless the UPS fails during a power failure. In this case, it is covered under Section 3.2 on critical failure procedures.

Failures in the components that supply and condition power to client components are not covered by this document.

2.5 Off-Site Software and Backup Storage Requirements

An off-site storage location is recommended to safeguard software installation media, documentation and backup tapes against being corrupted and/or destroyed in the event of an emergency which would render the PD^2 server inaccessible and/or damage its contents. The off-site storage location should be a locked physical space that can be accessed only by authorize a personnel. At a minimum, access to this space should be readily available to the PD^2 system administrator and the PD^2 database administrator 24 hours a day. This space should be conveniently located such that it can be accessed quickly in the event of an emergency but should not be located in the same area as the server.

3.0 Emergency Response Procedures

3.1 Non-Critical Failure of Hardware, Software or Network Component

ACTIVITY NAME SAs are responsible for recovery from failure of non-critical components. A non-critical failure does not affect availability of PD^2 database or application server capabilities that users depend on. For example, failure of the backup tape drive is a non-critical failure.

1. Assess impact of component failure(s).
   - Ensure that component failure is not critical to PD^2 operation and/or that it will not soon cause the failure of a critical system component.
   - Determine what (if any) disruptions in service will occur.

2. Notify management staff responsible for personnel affected by service disruptions.

3. Notify affected users of service disruptions (if any)

4. Determine if component can be repaired or replaced with, available staff and spare parts on-hand or if an external service organization must be contacted.

5. If required. contact external service organization responsible for the repair or replacement of failed component.

6. Schedule repair or replacement of component based on:

   • Availability of internal staff and/or staff from an external service organization,

   • Availability of spare or replacement parts, and

   • The need to minimize service disruptions.

7. Notify management and affected users of any additional service disruptions that will result from the maintenance activities required to repair or replace the failed component(s).

8. Perform scheduled repair and/or replacement of failed component(s).

9. Notify management and affected users that system operations have returned to normal.

10. Re-order and/or replenish spare parts and/or components used to repair or replace failed component(s).

3.2 Critical Failure of Hardware, Software or Network Component

ACTIVITY NAME SAs are responsible for recovery from failure of critical components. A critical component is one that adversely affects availability of PD^2 services to users. An example of a critical component failure is a disk drive failure that brings down the Unix Server machine, the DBMS (Sybase) or the PD^2 database.

1. Determine the full extent of service disruptions.

2. Notify management staff responsible for personnel affected by service disruptions.

3. Notify affected users of service disruptions.

4. Determine if component can be repaired or replaced with available staff and spare parts on hand or if an external service organization must be contacted.

5. If required, contact external service organization responsible for the repair or replacement o failed component.

6. Schedule repair or replacement of component immediately.

7. Notify management and affected users of any additional service disruptions that will result from the maintenance activities required to repair or replace the failed component(s), and inform them of approximate recovery time.

8. Perform repair and/or replacement of failed component(s).

9. Notify management and affected users that system operations have returned to normal.

10. Re-order and/or replenish spare parts and/or components used to repair or replace failed component(s).

3.3 Sudden Power Failure of Unknown Duration

1. Determine what PD^2 hardware and network components have been or could potentially be affected by the power.

2. Verify that the database and application servers have automatically powered down within 15 minutes of power outage.

3. If power to client workstations and printers has failed, notify all affected users to turn off the power-switches to the affected client workstations and printers.

4. Notify external service organization responsible for restoring power to the affected site(s).

5. Notify management staff responsible for personnel affected by service disruptions.

6. Notify affected users of service disruptions.

7. If power failure is estimated to take longer than 24 hours, determine if any backup operation response plans should be activated.

8. Notify management and affected users of any additional service disruptions that will result from the maintenance activities required to restore power to affected components.

10. Perform server startup procedures for all PID 2 server components.
11. Perform activities required to restore affected network routers to normal activity.
12. As necessary, turn affected client workstations and printers back on.
13. Inform users and other potentially affected entities of return to normal operations.

4.0 Backup Operations

4.1 Responsibilities

System administration services, including backup and recovery, for the database server is provided by ***ACTIVITY NAME*** SAs. This section is oriented to server backup operations. Specific responsibilities include the following activities:

- Ensure integrity of backup/recovery mechanisms at the site level (test procedures for restart/recovery, ensure off-site backups are kept, etc.).
- Maintain schedule of backup tasks that run at regular intervals.
- Monitor proper operation of backup background tasks.
- Provide first line of troubleshooting for problems relating to backup and recovery.

4.2 Database Server

For the database server, the System Administrator (SA) backs up the PD^2 database nightly. Backup tapes for the database are kept for two weeks before being recycled. Backup tapes are stored off-site in ***LOCATION OF OFF SITE TAPES***.

The dump file created by Sybase is stored in the local server directory; this file will then be backed up to tape by ***ACTIVITY NAME*** SAs after each dump has completed. This online dump file is over-written each time a backup is taken.

Backup operations commence at ***TIME BACK UP IS SCHEDULED***.

5.0 Recovery Operations

The two primary objectives of backup and recovery activities are to minimize the time required to fully recover PD^2 software and data files and minimize the amount of time that PD^2 is unavailable and/or operating at a reduced service capacity due to backup operations.

5.1 Database Server

For the database server, recovery of the database is of paramount importance. In event of failure of a single disk drive the SAs will recover from the prior day's backup and hourly syslog and will replace the failed drive with a new one. Recovery time will depend on the time it takes to install and reload data, but will not depend on the availability of a spare drive.

For failure of other components, such as a CPU chip, the ability of the database server to continue functioning effectively, and the time required to bring it back to full operational status. will depend on the exact nature of the failure. For example, for a -six CPU server, failure of a single CPU will reduce performance, but allow for continued operation of the server. In fact, with low load on the server, such a failure will be transparent to users.

Software failure or corruption is similar to failure of a non-disk drive component: operational status of the server, and recovery time will depend on the exact nature of the failure.

6.0 Maintenance Procedures

ACTIVITY NAME SAs are responsible for regularly scheduled maintenance activities, including those listed below:

- Maintain calendar of maintenance activities.
- Use DBCC (Database Consistency Checker) to check database consistency. See Appendix E for procedures on using DBCC.
- Examine the error log contents for SQL Server. Backup Server^{tm}, and SQL Monitor Server^{tm} regularly.
- Run the update statistics command regularly in order to ensure effective database performance.
- Examine auditing information.
- Recompile stored procedures, as required.
- Monitor the resource utilization of the server machine. This includes peak load CPU utilization, overall disk drive storage use vs. storage capacity, and relative disk drive spindle use for assessment of load leveling.

One key area of maintenance activities relates to monitoring usage of the database server. Key dimensions of server usage monitoring are described below:

Database Consistency Checker (DBCC). The DBCC is a set of utility commands used for checking the logical and physical consistency of a database. SAs should use DBCC commands as part of regular database maintenance. These checks can detect, and often correct, errors before they obstruct a user's ability to use the SQL Server. As a result, AMS recommends that the SA run DBCC before backing up any and all databases to ensure that the database is not corrupt. See Appendix E for instructions on using DBCC.

The SA should also run DBCC when a database is possibly damaged (such as after a system error has occurred or when messages in the error log or queries do not act as expected). If the SA finds that the database is damaged. the SA can run DBCC commands to determine the extent of damage. When dropping a database that is marked suspect, the SA must first run the DBCC commands.

Server Disk Space Allocation. If the PD^2 database is residing on its own dedicated server, the only issue regarding allocating disk space is the assurance that additional disk space can be provided.

Fine Tuning Sybase Memory Allocation. Memory is the most important configuration option. Setting this parameter incorrectly affects performance dramatically. To optimize the size of memory for NAVFACCO's system, a System Administrator calculates the memory required for the operating system and other uses and subtracts this from the total available physical memory. See Appendix F for specific memory allocation guidance.

Audit Features. PD^2 and Sybase come with built-in auditing features that provide an audit trail that can be used to detect misuse of system data and other resources.

Responsibility for Server Usage Monitoring. ***ACTIVITY NAME*** SAs are responsible for Database Monitoring Procedures to include the DBCC, disk space allocation, fine tuning Sybase memory allocation, and audit checking.

STANDARD PROCUREMENT SYSTEM
(SPS)

Continuity of Operations Plan
(COOP)

APPENDICES

Appendix A: Acronyms and Abbreviations

AMS	American Management Systems
CAO	Central Automation Office
CAPS	Computerized Accounts Payable System
CD	Compact Disc
CDA	Central Design Activity
COTS	Commercial Off -the-Shelf
CPU	Central Processing Unit
CST	Central Standard Time
DAT	Digital Audio Tape
DBA	Database Administrator
DBCC	Database Consistency Checker
DBMS	Database Management System, e.g., Sybase, Oracle
DFARS	Defense Federal Acquisition Supplement
DFAS	Defense Finance and Accounting Service
DOD	Department of Defense
EC	Electronic Commerce
EDI	Electronic Data Interchange
EST	Eastern Standard Time
FACSO	Facilities Systems Office
FAR	Federal Acquisition Regulation
FDDI	Fiber Distributed Data Interface
FOC	Full Operating Capability
FPDS	Federal Procurement Data System
GB	Gigabytes
GUI	Graphical User Interface
HQ	Headquarters
I/O	Input / Output
IA	Interface Agent
IAMS	Interface Administration System
IAW	In accordance with
IDB	Interface Database
IMD	Information Management Directorate
IRPRS	Integrated Requirements and Purchasing Request System
KB	Kilo-bytes
LAN	Local Area Network
MB	Megabytes
MHz	Megahertz
MONGOOSE	DOD Fraud Detection Operation
MS	Microsoft
MST	Mountain Standard Time
NT	New Technology
OAM	Object Allocation Map
PARC	Principal Assistant for Contracting
PC	Personal Computer
PD2	Procurement Desktop-Defense
PMO	Program Management Office

POC	Point-of-Contact
PST	Pacific Standard Time
RAID	Redundant Array of Inexpensive Devices
RAM	Random Access Memory
SA	System Administrator
SF	Standard Form
SPS	Standard Procurement System
TBD	To be determined
TCP/IP	Transmission Control Protocol / Internet Protocol
UPS	Uninterruptible Power Supply
WAN	Wide Area Network

Appendix B: Creating Database Devices

B.1 Creating a Database Device in SQL Central

In order to create a database device the diskinit command can be used in WISQL or the SQL Central tool may be used. The following section details how to use SQL Central to accomplish this task.

Using the SA login and password, login to SQL Server through SQL Central. Click "Database Devices" in the left-hand portion of the screen (see Figure B.1.1), and then double-click on the 'Add new database device" icon on the right hand portion of the screen.

Figure B.1.1 Adding a New Database Device

<INSERT SCREEN SHOT IF DESIRED>

The "Create New Database Device" dialogue box appears, and the option of creating a logical and explicit name for the new device is available. It would be a good idea to call the database device a name similar to the name of the database. Figure B.1.2 shows this window, and after clicking the "next" button, the option will be available to input the size of the device to be created (Figure B.1.3). Be aware of the database device location and make sure there enough free disk space to create a database on that drive.

Figure B.1.2 Create New Database Device Figure B.1.3 View after "Next" Button

<INSERT SCREENS SHOT IF DESIRED>

B.2 Creating a Database on a Logical Device

Again as with most Sybase tasks, either WISQL or SQL Central can be used to create the new database. In WISQL the command is *create database.* Syntax for this and other SQL commands can be found in the Sybase Manuals (SyBooks). From SQL Central, the process is very quick and easy.

After creating the database device where the new database will be placed, click on the database folder in SQL Central, and double-click on the "Add database" icon (see Figure B.2.1.).

Figure B.2.1. "Add Database" Icon

<INSERT SCREEN SHOT IF DESIRED>

The following set of dialogue boxes appears and offers a guide through the process of adding a new database to the SQL Server. (Figures B.2.2 and B.2.3).

Figure B.2.2 Specify Database Name Figure B.2.3 Specify Device Size

<INSERT SCREEN SHOTS IF DESIRED>

After naming the new database, it is then available on the list of database devices on the SQL Server. Then, the space occupied can be determined for the database device. Database size is limited to the size of the configured database device.

Figure B.2.4

<INSERT SCREEN SHOT IF DESIRED>

The newly created database can be seen in the Database view of SQL Central, as shown in Figure B.2.4 above. In order to view the properties of the new database (or to check the properties of any object in SQL Central) highlight the object in question right click, and choose properties from the resulting pop-up menu. The tab should display exactly what devices are being used for a specific database (see Figures B.2.5 and B.2.6). The option for "Truncate log on checkpoint" may be selected here as well. This option is enabled on all AMS provided databases.

Figure B.2.5 Associate Database with Device Figure B.2.6 Specify Options

<INSERT SCREEN SHOTS IF DESIRED>

Keep in mind that it is possible for a database to use more than one database device, thus making the process of increasing the size of any database very easy.

B.3 Growing a Database

In order to increase or "grow" the size of a database, another database device needs to be created. Once that process is done (as outlined in Appendix C), the newly created device can be added to the database. When clicking on the "add" button another dialogue box will appear to allow specification of the device to be added and space allocation. The amount of space included can be specified from any database device by clicking on the Edit option. This is shown in Figure B.3.2.

Figure B.3.1 Available Devices Figure B.3.2 Sizing the Device

<INSERT SCREEN SHOTS IF DESIRED>

B.4 Shrinking a Database

Shrinking a database is not recommended. For this reason, AMS has created databases that do not consume the server's entire drive space. It is advisable to start databases sizes small, and continue to maintain and enlarge them when needed.

B.5 Dump Devices

Another type of logical database device is a disk dump device. It is fairly simple to create a dump device and the database administrator must know how to create dump devices in order to backup databases.

From SQL Central, double-clicking the icon "Add Dump Device" under the Dump Device Folder creates a dump device. Dump devices are like database devices in that they must have a logical name and an exact physical location but differ in that they do not have size specified. For more information about creating dump devices and performing backups please see the section on backing up and restoring in this document.

Appendix C: Adding Space to the PD^2 Database

Use the following procedure to add space to the PD^2 database.

1. Through "SQL Central", log into the SQL Server as the system administrator (sa)

2. Add a new database device:

   - Select the "Database Device" folder
   - Double-click the "Add new database device" icon
   - Specify the name of the new device. *Naming convention* - when adding a new device to a database, should be the name of the database with the "_add" appendage. Each additional device added to this database should follow the same convention "_add1. _add2", etc.
   - Specify the full "path" for the new device (the *.dat file). Click "NEXT"
   - Specify the "device number" - the default should be ok, unless the number of devices has been exceeded
   - Specify the "size" of the device in MB
   - Ignore starting address. Click "NEXT"
   - Choose whether or not to mirror the new database device - probably not. Click "FINISH" to create the new device.

3. Extend an existing database with a new device:

   - Select the "Database" folder
   - Right-click the database that is to be extended and select "Properties"
   - Under the "Devices" tab, click the "Add" command button
   - Choose the type of device that is to be added to the database: choose "data"
   - Select the "database device" you want to add to the database

- Select the "amount of space" on that device to be allotted to the database. Click "OK" Click "Apply"

Appendix D: Database Procedures

This section describes a number of key database administration activities and procedures.

Backup Automation Procedures. Creating an automated backup procedure takes the guesswork out of performing backups and makes the procedure easier and quicker to perform. Automating backups can be as simple as using an operating system script or utility (for example, the UNIX cron utility) to perform the necessary backup commands or the procedure can be further automated using thresholds.

Although the commands required to create an automated script vary, depending on the operating system used, all scripts should accomplish the same basic steps:

- Use DBCC to verify the database's consistency
- Start WISQL and dump the database
- Rename the dump file to a name that contains the dump date, time, and database name
- Append a history file so that it includes the latest transaction
- Record any errors that occurred during the dump in a separate error file
- Alert the System Administrator as to any error conditions.

Making Routine Database Dumps. Dumping the database makes a copy of the entire database, including both the data and the transaction log. Dumping the database does not truncate the log. Users can continue to make changes to the database while the dump takes place. This feature allows a convenient back up of databases to occur on a regular basis.

The process of dumping the database executes in three phases. A progress message notes when each of the phases is completed. When the dump finishes, it reflects all changes made during its execution.

If the trunc log on chkpt database option is set to true, and the transaction log contains 50 or more rows, SQL Server automatically truncates the log when an automatic checkpoint occurs. Then, the entire database must be dumped (not just the transaction log) to ensure recoverability.

Backing Up the Master Database. Backups of the master database are used as part of the recovery procedure in case of a failure that affects the master database. If there is no current backup of master, vital system tables must be reconstructed at a time when it is a priority to get the databases up and running again. Be prepared to back up the master database regularly and frequently.

Back up the master database with dump database each time it is changed. Although the creation of database objects can be restricted in master, system procedures such as sp_addlogin and

sp_droplogin, sp_password and sp_modifylogin allow users to modify its system tables. Be sure to back up the master database on a frequent basis to record these changes.

Backing up the master database is the cornerstone of any backup and recovery plan. The master database contains details about the structure of the entire database system. Because SQL Server needs this information during recovery, it is crucial to maintain an up-to-date backup copy of the master database at all times.

To ensure that the backup of master is always up to date, back up the database after each command that affects disks, storage, databases, or segments. This means master should be backed up after performing any of the following procedures:

- Create or delete databases
- Initialize new database devices
- Add new dump devices
- Use any device mirroring command
- Create, drop, or modify a segment
- Add new SQL Server logins

The sybsystemprocs database stores only system procedures. It is very easy to restore this database by running the “install master script”, unless changes are made to the database. Sybsystemprocs should be backed up each time it is changed.

Truncating the Master Database Transaction Log. Since the master database transaction log is on the same database devices as the data, its transaction log cannot be backed up separately. The log of the master database cannot be moved. A dump database must be used to back up the master database. Use dump transaction with the “truncate_only” option periodically (for instance, after each database dump) to purge the transaction log of the master database.

Use Tapes for Dump Devices. Tapes are preferred as dump devices since they permit a library of database and transaction log dumps to be kept offline. Large databases can span multiple tape volumes. On UNIX systems, the Backup Server requires non-rewinding tape devices for all dumps and loads.

In general, dumping to a file or to a disk is not recommended. If the disk or computer containing that file crashes, there may be no way to recover the dumps. On UNIX and PC systems, dumping to a file or disk is the only option if the master database is too large to fit on a single tape volume, unless there is a second SQL Server that can issue sp_volchanged requests.

Dumps to a file or disk can be copied to tape for offline storage, but these tapes must be copied back to an online file before SQL Server can read them. The backup server cannot restore a database from a tape drive that contains a disk dump file that was manually backed up to the tape medium.

When backup scripts and threshold procedures are created, use logical names, rather than physical device names, whenever possible. Scripts and procedures that refer to actual device names must be modified each time a backup device is replaced. If scripts and procedures refer to logical device names, the sysdevices entry can simply be dropped for the failed device and a new entry will associate the logical name with a different physical device.

Once backup procedures have been developed and tested, commit them to paper. Determine a reasonable backup schedule and adhere to it. If backup procedures are developed, documented and tested ahead of time, it will be easier to get databases online when disaster strikes.

Disk Mirroring. Disk mirroring can provide nonstop recovery in the event of media failure. The disk mirror command causes a SQL Server database device to be duplicated - all writes to the device are also written to a separate physical device. If one of the devices fails, the other contains an up-to-date copy of all transactions.

When a read or write to a mirrored device fails, SQL Server "unmirrors" the bad device and displays error messages. SQL Server continues to run unmirrored. To restart mirroring, the System Administrator must issue the disk remirror command. In order to decide exactly what devices to mirror please see the Sybase System Administration Guide.

Disk mirroring is available on the hardware level, the operating system level and the Sybase SQL Server level. There are many advantages to mirroring solution to data loss.

Using RAID. Mirroring disks on the operating system level is handled on many levels. The most common OS solution to data replication is the RAID scenario. RAID (redundant array of inexpensive devices) is typically made up of five devices; four with data, and one with parity information. In this system, failure of a single drive in the array will result in no loss of data. When the failed device is replaced, RAID automatically formats and populates the new device.

There are many levels of RAID technology offering varying levels of performance and reliability.

Level 0

- Performance only, no replication
- Data striping across multiple disks without providing for data redundancy
- High I/0 rate due to small block size

Level 1

- Traditional hardware level disk mirroring with no data striping
- Often this method provides a faster and more efficient mirroring mechanism than SQL Server software mirroring.

Level 5

- Full replication
- Independent array with data striping
- Does not maintain a single parity disk
- Parity information from one disk is spread across the other disks in the array
- SQL Server Level Guarantees database stability against hardware failure
- Enables mirroring across controllers as well as physical devices

Appendix E: Database Consistency Checker Procedures

Using the DBCC commands, System Administrators can check a database for errors before backing it up. DBCC commands verify that the storage of a database or database object “makes sense" to SQL Server. Always use DBCC commands to verify the integrity of a database before dumping it. If DBCC detects errors, correct them before dumping the database. Sybase recommends that the following DBCC commands be performed before backing up (dumping) the database:

checkcatalog: the DBCC checkcatalog command checks for consistency problems between system tables and within specific system tables in a database.
checkdb: the *DBCC checkdb* command performs a DBCC *checktable* command on each of the tables within a database.
checktable: the *DBCC checktable* command verifies page linkages, pointers, and indexes within a table. It also makes sure that data rows on each page have entries in an object allocation map (OAM) page.
checkalloc: the DBCC checkalloc command checks to see if the page allocation for a database is consistent. It checks to see that all pages have been correctly allocated, that no page is allocated that is not part of a page linkage, and vice versa. It also checks all allocation pages in a database to insure they contain valid information.

In general, the more thoroughly the DBCC command checks the integrity of an object or database, the slower it is. In order for the DBCC to validate the table allocation pages and page linkages it performs a huge amount of physical I/0 and locks tables from update activity while running. As a result, the process can be very time consuming and have a huge impact on on-line performance.

Over time, ***ACTIVITY NAME*** can begin to think of running DBCC as insurance for its databases. If a few errors are discovered or no errors while running DBCC in the past, ***ACTIVITY NAME*** may decide that the risk of database corruption is small and that DBCC can be run only occasionally. Or, if the consequences of losing data are too high, ***ACTIVITY NAME*** should continue to run DBCC commands before a database is backed up.

Appendix F: Memory Allocation Procedures

This section describes how to allocate RAM for the SQL Server product so that SQL Server takes full advantage of available CPU resources.

The total memory parameter sets the size of memory (in 2K units) that SQL Server allocates from the operating system. The more memory available, the more resources SQL Server has for internal buffers and caches, reducing the number of times the server has to read data from disk for static information or compiled procedure plans. There is no performance penalty for configuring SQL Server to use the maximum memory available to it. If the PD^2 database server is ever used to host other applications, NAVFACCO must re-assess memory allocation to ensure that SQL Server is able to acquire enough memory to boot.

The Sybase memory setting will be configured as part of SQL Server installation or upgrade. ***ACTIVITY NAME*** should review the total memory configuration parameter setting when the amount of RAM is changed; when the pattern of use of the machine changes; or if memory is allocated for extent I/0 buffers or additional network memory for SQL Server.

Directions:

- In SQL Central, login as “sa"
- Right click on the server on the left hand side of the screen and choose **Configure**
- Scroll down to **total memory** and in the value column put the new value (see how to choose the value below). **Remember that the value is entered in as 2K pages. That means that 32MB of RAM equals 16000 into the value column.**
- Click **OK**
- Reboot for the change to take effect.

Choosing the Memory Value. After determining the server operating system type, the total RAM in the server, and whether the server is a dedicated Sybase server, consult Table F-1 below. Caution: never set the Sybase memory value larger than the physical memory on the server - the Sybase server may not boot up. The most aggressive value is: (real memory) - *32MB* = memory value. Always leave at least 32MB of RAM for the operating system.

Table F-1 Setting Memory for SQL Server

All platforms require a minimum of 15MB (or 7500 2K pages) of RAM to boot SQL Server (Sybase SQL Server 11.0.2 or greater)

Server Type	Dedicated	RAM	Memory Setting
NT	Yes	64 MB	32 MB or 16000 2K pages
NT	Yes	128 MB	96 MB or 48000 2K pages
NT	No	Less than 128 MB	No change
NT	No	128 MB or more	32 MB or 16000 2K pages
Novell	No		48 MB or 24000 2K pages
UNIX	No	64 MB or more	Not greater than (TOTAL MB – 32MB or ((TOTAL – 32) * 512) 2K pages of **SHARED MEMORY**
UNIX	Yes	64 MB or more	Not greater than (TOTAL MB – 48MB or ((TOTAL – 48) * 512) 2K pages of **SHARED MEMORY**